

# The Grid File I/O (Gfio) Library

## Version 1.00

ROB LUCCHESI AND ARLINDO DA SILVA

*Data Assimilation Office, NASA/GSFC, Greenbelt, MD 20771*

August 3, 1998 (Original design October 1997)

## Contents

<b>1 System Overview</b>	<b>2</b>
<b>2 Routine/Function Prologues</b>	<b>4</b>
2.1 Gfio_Create — Creates a DAO gridded file for writing (Source File: gfio.F)	4
2.2 Gfio_Open — Opens an existing DAO gridded file (Source File: gfio.F) . . .	9
2.3 Gfio_PutVar — Writes variable to file at this time (Source File: gfio.F) . . .	10
2.4 Gfio_GetVar — Reads variable from file (Source File: gfio.F) . . . . .	16
2.5 Gfio_DimInquire — Gets dimension information from a GFIO (Source File: gfio.F) . . . . .	21
2.6 Gfio_Inquire — Get information about a GFIO file. (Source File: gfio.F) . .	23
2.7 Gfio_Close — Closes file (Source File: gfio.F) . . . . .	30
2.8 Gfio_PutIntAtt — Write a user-defined integer attribute (Source File: gfio.F)	31
2.9 Gfio_PutRealAtt — Write a user-defined real attribute (Source File: gfio.F)	32
2.10 Gfio_PutCharAtt — Write a user-defined character attribute (Source File: gfio.F) . . . . .	33
2.11 Gfio_GetAttNames - Get global attribute names (Source File: gfio.F) . . .	34
2.12 Gfio_AttInquire — Get information about an attribute (Source File: gfio.F)	36
2.13 Gfio_GetIntAtt — Write a user-defined integer attribute (Source File: gfio.F)	37
2.14 Gfio_GetRealAtt — Read a user-defined real attribute (Source File: gfio.F)	38
2.15 Gfio_GetCharAtt — Read a user-defined character attribute (Source File: gfio.F) . . . . .	39

## 1 System Overview

Basic Requirements:

- 
- (1) Design an interface for writing HDF format data files from the June 1998 GEOS-3 production system without requiring the direct insertion of HDF Toolkit calls in the code.
  - (2) Design this interface to be flexible enough to support usage in other DAO applications that read or write HDF data.
  - (3) Output files should conform the COARDS conventions. This allows the data to be immediately usable by GrADS, other visualization packages and utilites such as ncdump.
  - (4) Provide a library that is callable from a Fortran 77 application with a portable interface.
  - (5) The library must also be callable by C, perhaps with the use of a tool like Cfortran.h.

The primary motivation behind GFIO is to provide an easy way for the GEOS-DAS to write HDF format data while hiding calls to the HDF libraries. Additionally, it is hoped that this library will be of general use for reading or writing HDF files in applications other than the GEOS-DAS.

The typical calling sequence for creating a file would be:

```
GFIO_Create(...)  
  
GFIO_PutVar(...)  
GFIO_PutVar(...)  
GFIO_PutVar(...)  
. . .  
GFIO_Close(...)
```

One could subsequently open the file for more writing with:

```
GFIO_Open(...)
```

NOTES:

-----

- \* Surface data is permitted in the same file as upper air data, however all upper air data must be defined with the same number of levels but it is not necessary to write data for each defined level. In the case that data is not written for a given level, HDF will put fills.
- \* Packing is not yet implemented.
- \* The time increment cannot be defined as 0, even if only writing one time.
- \* Files are written using the NetCDF interface provided in the HDF library. The files conform to the COARDS conventions, meaning they have specific metadata defined by the convention.

## 2 Routine/Function Prologues

### 2.1 Gfio\_Create — Creates a DAO gridded file for writing (Source File: *gfio.F*)

This routine is used to open a new file for a GFIO stream. Packing is not yet supported. Information about each opened stream is stored in a COMMON block contained in *gfio.h*. This information is later used by GFIO\_PutVar. GFIO\_OpenW should be used to open an existing file for writing.

**INTERFACE:**

```
subroutine Gfio_Create ( fname, title, source, contact, amiss,
&                               im, jm, km, lon, lat, levs, levunits,
&                               yyyyymmdd_beg, hhmmss_beg, timinc,
&                               nvars, vname, vtitle, vunits, kmvar,
&                               valid_range, packing_range,
&                               fid, rc )
```

**USES:**

```
Implicit NONE
include 'netcdf.inc'
include 'gfio.h'
```

**INPUT PARAMETERS:**

		! ----- Global Metadata -----
character(*)	fname	! File name
character(*)	title	! A title for the data set
character(*)	source	! Where it came from
character(*)	contact	! Who to contact about the data set, e.g., ! 'Contact data@dao.gsfc.nasa.gov'
real	amiss	! Missing value such as 1.0E15
		! ----- Dimension Metadata -----
integer	im	! size of longitudinal dimension
integer	jm	! size of latitudinal dimension
integer	km	! size of vertical dimension
real	lon(im)	! longitude (in degrees ?????) of center ! of gridbox
real	lat(jm)	! latitude (in degrees north) of center ! of gridbox
real	levs(km)	! Level (units given by levunits) of ! center of gridbox
character(*)	levunits	! units of level dimension, e.g., ! "hPa", "sigma_level"
integer	yyyyymmdd_beg	! First year-month-day to be written
integer	hhmmss_beg	! First hour-minute-second to be written
integer	timinc	! Increment between output times (HHMMSS)

```

                                ! ----- Variable Metadata -----
integer      nvars          ! number of variables on file
character(*) vname(nvars)   ! variable short name, e.g., "hght"
character(*) vtitle(nvars)  ! variable long name, e.g.,
                            !    "Geopotential Height"
character(*) vunits(nvars)  ! variable units, e.g., "meter/second"
integer       kmvar(nvars)   ! number of levels for variable; it can
                            ! either be 0 (2-D fields) or equal to km
real         valid_range(2) ! optional variable valid range; set to
                            !    "amiss" if not known.

                                ! ----- Optional Packing Metadata ---
real         packing_range(2) ! optional variable range to be used
                            ! for 16-bit packing. If packing is
                            ! not desired, just set it to "amiss".
! NOTE:
! The packing algorithm sets all values
! outside the packing range to missing.

```

*OUTPUT PARAMETERS:*

```

integer      fid            ! File handle
integer      rc             ! Error return code:
                            !   rc = 0    all is well
                            !   rc = -1   time increment is 0

```

*REVISION HISTORY:*

1997.09.13	da Silva/Lucchesi	Initial interface design.
1997.09.22	Lucchesi	Added timinc to interface.
1998.02.10	Lucchesi	Added support for applications running with 64-bit reals.
1998.02.17	Lucchesi	Added time_inc, begin_time, and begin_date attributes to the time dimension.
1998.03.30	Lucchesi	Documentation expanded. Clean-up of code.
1998.07.07	Lucchesi	Removed vids from argument list
1998.07.09	Lucchesi	Converted timinc to seconds before saving

*CONTENTS:*


---

! REAL\*4 variables for 32-bit output to netCDF file.

```

real*4 packing_range_32(2),valid_range_32(2)
real*4 amiss_32
real*4 lon_32(im), lat_32(jm), levs_32(km)
real*4 scale_32

```

```

real*4 offset_32
integer vid(nvars)
integer iret, i
integer timeid, latid, lonid, levid
integer timedim, latdim, londim, levdim
integer dims3D(4), dims2D(3)
integer corner(4), edges(4)
character*80 timeUnits
logical first
data first /.true./
character*8 strBuf
character*14 dateString
integer year,mon,day,hour,min,sec

! Basic error-checking.

if (timinc .eq. 0) then
  rc=-1
  return
endif

! Convert double-precision output variables to single-precision

do i=1,im
  lon_32(i) = lon(i)
enddo
do i=1,jm
  lat_32(i) = lat(i)
enddo
do i=1,km
  levs_32(i) = levs(i)
enddo
do i=1,2
  valid_range_32(i) = valid_range(i)
  packing_range_32(i) = packing_range(i)
enddo

amiss_32 = amiss

! Convert time increment to seconds.

      write (strBuf,203) timinc
203  format (I6)
      read (strBuf,204) hour, min, sec
204  format (3I2)
      timinc = hour*3600 + min*60 + sec

! Make NetCDF errors non-fatal, but issue warning messages.

```

```

call ncpopt(NCVERBOS)

! Create the new NetCDF file. [ Enter define mode. ]

fid = nccre (fname, NCCLLOB, iret)

! Define dimensions.

londim = ncddef (fid, 'lon', im, iret)
latdim = ncddef (fid, 'lat', jm, iret)
levdim = ncddef (fid, 'lev', km, iret)
timedim = ncddef(fid, 'time', NCUNLIM, iret)

! Define dimension variables.

lonid = ncvdef (fid, 'lon', NCFLOAT, 1, londim, iret)
latid = ncvdef (fid, 'lat', NCFLOAT, 1, latdim, iret)
levid = ncvdef (fid, 'lev', NCFLOAT, 1, levdim, iret)
timeid = ncvdef (fid, 'time', NCLONG, 1, timedim, iret)

! Set attributes for dimensions.

call ncaptc (fid,lonid,'long_name',NCCHAR,9,'longitude',iret)
call ncaptc (fid,lonid,'units',NCCHAR,12,'degrees_east',iret)

call ncaptc (fid,latid,'long_name',NCCHAR,8,'latitude',iret)
call ncaptc (fid,latid,'units',NCCHAR,13,'degrees_north',iret)

call ncaptc (fid,levid,'long_name',NCCHAR,14,'vertical level',
.           iret)
call ncaptc (fid,levid,'units',NCCHAR,LEN(levunits),levunits,iret)

call ncaptc (fid, timeid, 'long_name', NCCHAR, 4, 'time', iret)

write (dateString,200) yyyyymmdd_beg, hhmmss_beg
200 format (I8,I6)
read (dateString,201) year,mon,day,hour,min,sec
201 format (I4,5I2)
write (timeUnits,202) year,mon,day,hour,min,sec
202 format ('minutes since ',I4.4,'-',I2.2,'-',I2.2,' ',I2.2,':',
.           I2.2,':',I2.2)
call ncaptc (fid, timeid, 'units', NCCHAR, 33, timeUnits, iret)
call ncapt (fid, timeid, 'time_increment', NCLONG, 1, timinc,
.           iret)
call ncapt (fid,timeid,'begin_date',NCLONG,1,yyyyymmdd_beg,iret)
call ncapt (fid,timeid,'begin_time',NCLONG,1,hhmmss_beg,iret)

```

```

dims3D(4) = timedim
dims3D(3) = levdim
dims3D(2) = latdim
dims3D(1) = londim

dims2D(3) = timedim
dims2D(2) = latdim
dims2D(1) = londim

scale_32 = 1.0      ! No packing for now.
offset_32 = 0.0      ! No packing for now.

! Define physical variables. Set attributes for physical variables.

do i=1,nvars
  if ( kmvar(i) .eq. 0 ) then
    vid(i) = ncvdef (fid, vname(i), NCFLOAT, 3, dims2D, iret)
  else
    vid(i) = ncvdef (fid, vname(i), NCFLOAT, 4, dims3D, iret)
  endif
  call ncaptc (fid, vid(i), 'long_name', NCCHAR, LEN(vtitle(i)),
.           vtitle(i), iret)
  call ncaptc (fid, vid(i), 'units', NCCHAR, LEN(vunits(i)),
.           vunits(i), iret)
  call ncapt (fid,vid(i),'missing_value',NCFLOAT,1,amiss_32,iret)
  call ncapt (fid,vid(i),'scale_factor',NCFLOAT,1,scale_32,iret)
  call ncapt (fid,vid(i),'add_offset',NCFLOAT,1,offset_32,iret)
enddo

! Define global file attributes.

call ncaptc (fid,NCGLOBAL,'Conventions',NCCHAR,6,'COARDS',iret)
call ncaptc (fid,NCGLOBAL,'Source',NCCHAR,LEN(source),source,iret)
call ncaptc (fid,NCGLOBAL,'Title',NCCHAR,LEN(title),title,iret)
call ncaptc (fid,NCGLOBAL,'Contact',NCCHAR,LEN(contact),contact,
.           iret)
call ncaptc (fid,NCGLOBAL,'History',NCCHAR,10,'No History',iret)

! Exit define mode.

call ncendf (fid, iret)

! Write out dimension variables.

corner(1) = 1
edges(1) = im
call ncvpt (fid, lonid, corner, edges, lon_32, iret)

```

```

corner(1) = 1
edges(1) = jm
call ncvpt (fid, latid, corner, edges, lat_32, iret)

corner(1) = 1
edges(1) = km
call ncvpt (fid, levid, corner, edges, levs_32, iret)

corner(1) = 1
edges(1) = 1
call ncvpt (fid, timeid, corner, edges, 0, iret)

rc=0
return
end

!-----  

!      NASA/GSFC, Data Assimilation Office, Code 910.3, GEOS/DAS      !
!-----  


```

---

## 2.2 Gfio\_Open — Opens an existing DAO gridded file (Source File: *gfio.F*)

This routine opens an existing DAO gridded file. The file mode will be read/write. If the application already knows the contents of the file, it may begin interaction with the file using the returned file handle. Otherwise, the file handle can be used with the "inquire" routines to gather information about the contents. A negative return code indicates there were problems opening the file.

### INTERFACE:

```
subroutine Gfio_Open ( fname, fid, rc )
```

### USES:

```
Implicit NONE
include 'netcdf.inc'
include 'gfio.h'
```

### INPUT PARAMETERS:

```
character(*)    fname          ! File name
```

### OUTPUT PARAMETERS:

```

integer      fid          ! File handle
integer      rc           ! Error return code:
                  !   rc = 0    All is well
                  !   rc < 0   Error opening file

```

**REVISION HISTORY:**

1998.07.02	Lucchesi	Initial interface design.
1998.07.07	Lucchesi	Initial coding.

**CONTENTS:**


---

```

!-----
      fid = ncopn (fname, NCWRITE, rc)
      if ( rc .GT. 0) then
          rc = -1
      endif
      return
      end

!-----
!       NASA/GSFC, Data Assimilation Office, Code 910.3, GEOS/DAS      !
!-----
```

---

**2.3 Gfio\_PutVar — Writes variable to file at this time (Source File: *gfio.F*)**

This routine is used to write a variable to an open GFIO stream. Multiple vertical levels can be written at one time provided they are contiguous in memory. Date and time must be consistent with the time increment and the starting date/time as defined in GFIO\_Create. Times must fall on minute boundaries to allow GrADS to work. Error checking is done for dimensions that are out of bounds.

**INTERFACE:**

```

subroutine Gfio_PutVar ( fid, vname, yyyyymmdd, hhmmss,
&                         im, jm, kbeg, kount, grid,
&                         rc )

```

**USES:**

```

Implicit NONE
include 'netcdf.inc'
include 'gfio.h'
```

**INPUT PARAMETERS:**

```

integer      fid          ! File handle
character*(*) vname        ! Variable name
integer      yyyyymmdd    ! Year-month-day, e.g., 19971003
integer      hhmmss       ! Hour-minute-second, e.g., 120000

integer      im           ! size of longitudinal dimension
integer      jm           ! size of latitudinal dimension
integer      kbeg         ! first level to write; if 2-D grid
                      ! use kbeg = 0.
integer      kount        ! number of levels to write
real        grid(im,jm,kount) ! Gridded data to write at this time

```

*OUTPUT PARAMETERS:*

```

integer      rc   ! Error return code:
               ! rc = 0 all is well
               ! rc = -1 invalid fid
               ! rc = -2 time not consistent with increment
               ! rc = -3 problem with level
               ! rc = -4 problem with im or jm
               ! rc = -5 time does not fall on a minute boundary
               ! rc = -6 an error occurred in diffdate
               ! rc = -7 vname not defined
               ! rc = -8 time information not in metadata
               ! rc = -9 error writing data to file

```

*REMARKS:*

## Design issues:

1. What is the best way to represent time?
2. Should we force the user to write the full 3-D array? This improves efficiency and simplifies interface.

*REVISION HISTORY:*

1997.10.13	da Silva/Lucchesi	Initial interface design.
1998.02.10	Lucchesi	Added support for applications running with 64-bit reals.
1998.03.30	Lucchesi	Documentation expanded. Clean-up of code.
1998.07.02	Lucchesi	Replaced vid with vname in argument list & made related mods to code.

*CONTENTS:*


---

```
#if defined (REAL8)
  real*4 grid_32(im,jm,kount)
```

```

#endif

#if defined (DIM_CHECK)
    integer timeid, dimSize, dimId
    character*MAXCHR dimName
#endif

        integer corner(4), edges(4)
        integer vid
        integer iret
        integer seconds, DiffDate, timeIndex
        integer minutes           ! added as a work-around
        integer idx, i, j, k
        integer begDate, begTime, timInc

! Make NetCDF errors non-fatal, but issue warning messages.

call ncpopt(NCVERBOS)

#if defined (DIM_CHECK)

! Check to make sure max string lengths are large enough.  NetCDF defines
! MAXNCNAM, but it can't be used in a character*MAXNCNAM statement.
! MAXCHR is a CPP define in the gfoio.h file.

    if (MAXCHR .LT. MAXNCNAM) then
        print *, 'GFOIO_PutVar warning: MAXNCNAM is larger than ',
        .          'dimName array size.'
    endif
#endif

! Determine NetCDF variable ID.

    vid = ncvid (fid, vname, iret)
    if ( iret .NE. 0 ) then
        rc = -7
        return
    endif

! Basic error checking

#if defined (DIM_CHECK)
    dimId = ncidid (fid, 'lon', rc)
    if (rc .NE. 0) then
        rc = -4
        return
    endif
    call ncdinq (fid, dimId, dimName, dimSize, rc)

```

```

if (rc .NE. 0) then
  rc = -4
  return
endif
if (dimSize .ne. im) then
  rc = -4
  return
endif

dimId = ncdid (fid, 'lat', rc)
if (rc .NE. 0) then
  rc = -4
  return
endif
call ncdinq (fid, dimId, dimName, dimSize, rc)
if (rc .NE. 0) then
  rc = -4
  return
endif
if (dimSize .ne. jm) then
  rc = -4
  return
endif

dimId = ncdid (fid, 'lev', rc)
if (rc .NE. 0) then
  rc = -3
  return
endif
call ncdinq (fid, dimId, dimName, dimSize, rc)
if (rc .NE. 0) then
  rc = -3
  return
endif
if (kbeg-1 + kount .gt. dimSize) then
  rc = -3
  return
endif
#endif

! Determine number of seconds since starting date/time.

timeId = ncvid (fid, 'time', rc)
if (rc .NE. 0) then
  rc = -8
  return
endif
call ncagt (fid, timeId, 'begin_date', begDate, rc)

```

```

if (rc .NE. 0) then
  rc = -8
  return
endif
call ncagt (fid, timeId, 'begin_time', begTime, rc)
if (rc .NE. 0) then
  rc = -8
  return
endif
seconds = DiffDate (begDate, begTime, yyyyymmdd, hhmmss)

if (seconds .lt. 0) then
  print *, 'GFTIO_putvar: Error code from diffdate. Problem with',
  ' date/time.'
  rc = -6
  return
endif
if ( MOD (seconds,60) .eq. 0 ) then
  minutes = seconds / 60
else
  print *, 'GFTIO_putvar: Currently, times must fall on minute ',
  'boundaries.'
  rc = -5
  return
endif

! Confirm that this time is consistent with the starting time coupled with
! the time increment.

call ncagt (fid, timeId, 'time_increment', timInc, rc)
if (rc .NE. 0) then
  rc = -8
  return
endif

if ( MOD (seconds, timInc) .ne. 0 ) then
  print *, 'GFTIO_putvar: Absolute time of ',seconds,' not ',
  'possible with an interval of ',timInc
  rc = -2
  return
else
  timeIndex = seconds/timInc + 1
endif

#if defined (REAL8)

! Convert double-precision grid to single-precision for output

```

```

do k=1,kount
  do j=1,jm
    do i=1,im
      grid_32(i,j,k) = grid(i,j,k)
    enddo
  enddo
enddo

#endiff

! Load starting indicies.

if ( kbeg .eq. 0 ) then
  corner(1)=1
  corner(2)=1
  corner(3)=timeIndex
  edges(1)=im
  edges(2)=jm
  edges(3)=1
else
  corner(1)=1
  corner(2)=1
  corner(3)=kbeg
  corner(4)=timeIndex
  edges(1)=im
  edges(2)=jm
  edges(3)=kount
  edges(4)=1
endif

#if defined (REAL8)
  call ncvpt (fid, vid, corner, edges, grid_32, rc)
#else
  call ncvpt (fid, vid, corner, edges, grid, rc)
#endiff

if (rc .NE. 0) then
  rc = -9
  return
endif

corner(1)=timeIndex
edges(1)=1
call ncvpt (fid,timeId,corner,edges,minutes,iret)
rc=0
return
end

```

---

```
!-----  
!      NASA/GSFC, Data Assimilation Office, Code 910.3, GEOS/DAS      !  
!-----
```

---

## 2.4 Gfio\_GetVar — Reads variable from file (Source File: *gfio.F*)

This routine will read one or more levels of "vname" into the buffer passed in as "grid." "fid" is the file handle returned by Gfio\_open.

*INTERFACE:*

```
subroutine Gfio_GetVar ( fid, vname, yyyyymmdd, hhmmss,  
&                      im, jm, kbeg, kount, grid, rc )
```

*USES:*

```
Implicit NONE  
include 'netcdf.inc'  
include 'gfio.h'
```

*INPUT PARAMETERS:*

integer	fid	! File handle
character(*)	vname	! Variable name
integer	yyyyymmdd	! Year-month-day, e.g., 19971003
integer	hhmmss	! Hour-minute-second, e.g., 120000
integer	im	! size of longitudinal dimension
integer	jm	! size of latitudinal dimension
integer	kbeg	! first level to read; if 2-D grid ! set kbeg = 0.
integer	kount	! number of levels to read

*OUTPUT PARAMETERS:*

real	grid(im,jm,kount)	! Gridded data read for this time
integer	rc	! Error return code: ! rc = 0 all is well ! rc = -1 time increment not found in file ! rc = -2 time not consistent with increment ! rc = -3 problem with level ! rc = -4 problem with im or jm ! rc = -5 time not on minute boundary ! rc = -6 error calculation time offset ! rc = -7 variable not found ! rc = -8 error getting begin date/time ! rc = -9 read of variable failed

## REVISION HISTORY:

1997.10.13 da Silva/Lucchesi	Initial interface design.
1998.07.07 Lucchesi	Combined two GetVar routines into this one.

## CONTENTS:

```
!-----
```

```

integer timeId, begDate, begTime, seconds, minutes, timInc
integer corner(4), edges(4), timeIndex
integer vid
integer DiffDate

#if defined (REAL8)
    real*4 grid_32(im,jm,kount)
#endif

#if defined (DIM_CHECK)
    integer dimSize, dimId
    character*MAXCHR dimName
#endif

! Make NetCDF errors non-fatal, but issue warning messages.

call ncpopt(NCVERBOS)

#if defined (DIM_CHECK)

! Check to make sure max string lengths are large enough.  NetCDF defines
! MAXNCNAM, but it can't be used in a character*MAXNCNAM statement.
! MAXCHR is a CPP define in the gfio.h file.

    if (MAXCHR .LT. MAXNCNAM) then
        print *, 'GFIO_GetVar warning: MAXNCNAM is larger than ',
        .      'dimName array size.'
    endif
#endif

! Determine NetCDF variable ID.

vid = ncvid (fid, vname, rc)
if ( rc .NE. 0 ) then
    rc = -7
    return
endif

#if defined (DIM_CHECK)
```

```
! Optional dimension checks.
```

```

dimId = ncdid (fid, 'lon', rc)
if (rc .NE. 0) then
  rc = -4
  return
endif
call ncdinq (fid, dimId, dimName, dimSize, rc)
if (rc .NE. 0) then
  rc = -4
  return
endif
if (dimSize .ne. im) then
  rc = -4
  return
endif

dimId = ncdid (fid, 'lat', rc)
if (rc .NE. 0) then
  rc = -4
  return
endif
call ncdinq (fid, dimId, dimName, dimSize, rc)
if (rc .NE. 0) then
  rc = -4
  return
endif
if (dimSize .ne. jm) then
  rc = -4
  return
endif

dimId = ncdid (fid, 'lev', rc)
if (rc .NE. 0) then
  rc = -3
  return
endif
call ncdinq (fid, dimId, dimName, dimSize, rc)
if (rc .NE. 0) then
  rc = -3
  return
endif
if (kbeg-1 + kount .gt. dimSize) then
  rc = -3
  return
endif
#endif
#endif
#endif
```

```
! Get beginning time & date. Calculate offset seconds from start.
```

```
timeId = ncvid (fid, 'time', rc)
if (rc .NE. 0) then
  rc = -8
  return
endif
call ncagt (fid, timeId, 'begin_date', begDate, rc)
if (rc .NE. 0) then
  rc = -8
  return
endif
call ncagt (fid, timeId, 'begin_time', begTime, rc)
if (rc .NE. 0) then
  rc = -8
  return
endif
seconds = DiffDate (begDate, begTime, yyyyymmdd, hhmmss)
```

```
! Make sure input time are valid.
```

```
if (seconds .lt. 0) then
  print *, 'GFIO_getvar: Error code from diffdate. Problem with',
  .      ' date/time.'
  rc = -6
  return
endif
if ( MOD (seconds,60) .eq. 0 ) then
  minutes = seconds / 60
else
  print *, 'GFIO_getvar: Currently, times must fall on minute ',
  .      'boundaries.'
  rc = -5
  return
endif
```

```
! Determine the time index from the offset and time increment.
```

```
call ncagt (fid, timeId, 'time_increment', timInc, rc)
if (rc .NE. 0) then
  rc = -1
  return
endif

if ( MOD (seconds, timInc) .ne. 0 ) then
  print *, 'GFIO_getvar: Absolute time of ',seconds,' not ',
  .      'possible with an interval of ',timInc
  rc = -2
```

```

        return
    else
        timeIndex = seconds/timInc + 1
    endif

! Load starting indicies.

if ( kbeg .eq. 0 ) then
    corner(1)=1
    corner(2)=1
    corner(3)=timeIndex
    edges(1)=im
    edges(2)=jm
    edges(3)=1
else
    corner(1)=1
    corner(2)=1
    corner(3)=kbeg
    corner(4)=timeIndex
    edges(1)=im
    edges(2)=jm
    edges(3)=kount
    edges(4)=1
endif

! Read data.

#if defined (REAL8)
    call ncvgt (fid, vid, corner, edges, grid_32, rc)
#else
    call ncvgt (fid, vid, corner, edges, grid, rc)
#endif

if (rc .NE. 0) then
    rc = -9
    return
endif

#if defined (REAL8)

! Convert single-precision input grid to double-precision.

do k=1,kount
    do j=1,jm
        do i=1,im
            grid(i,j,k) = grid_32(i,j,k)
        enddo
    enddo

```

```

    enddo
#endif

    return
end

!--- NASA/GSFC, Data Assimilation Office, Code 910.3, GEOS/DAS !!
!---

```

---

## 2.5 Gfio\_DimInquire — Gets dimension information from a GFIO (Source File: *gfio.F*)

file. This routine is used to get dimension information from an existing GFIO file. This dimension information can subsequently be used to allocate arrays for reading data from the file. For more complete information about the contents of a file, Gfio\_Inquire should be used.

### INTERFACE:

```
subroutine Gfio_DimInquire (fid,im,jm,km,lm,nvars,ngatts,rc)
```

### USES:

```
Implicit NONE
include 'netcdf.inc'
include 'gfio.h'
```

### INPUT PARAMETERS:

integer	fid	! File handle
---------	-----	---------------

### OUTPUT PARAMETERS:

integer	im	! Size of longitudinal dimension
integer	jm	! Size of latitudinal dimension
integer	km	! Size of vertical dimension
integer	lm	! Number of times
integer	nvars	! Number of variables
integer	ngatts	! Number of global attributes
integer	rc	! Error return code:
! rc = 0 all is well ! rc = -1 invalid fid or missing dimension		

### REVISION HISTORY:

1998.07.02	Lucchesi	Initial interface design.
1998.08.05	Lucchesi	Added "ngatts"

## CONTENTS:

---

```

integer timeid, dimId
character*MAXCHR dimName
integer nDims

! Make NetCDF errors non-fatal, but issue warning messages.

call ncpopt(NCVERBOS)

! Check FID here.

! Check to make sure max string lengths are large enough. NetCDF defines
! MAXNCNAM, but it can't be used in a character*MAXNCNAM statement.
! MAXCHR is a CPP define in the gfio.h file.

if (MAXCHR .LT. MAXNCNAM) then
    print *, 'GFIO_PutVar warning: MAXNCNAM is larger than ',
    .           'dimName array size.'
endif

! Extract dimension information

dimId = ncdid (fid, 'lon', rc)
if (rc .NE. 0) then
    rc = -1
    return
endif
call ncdinq (fid, dimId, dimName, im, rc)
if (rc .NE. 0) then
    rc = -1
    return
endif

dimId = ncdid (fid, 'lat', rc)
if (rc .NE. 0) then
    rc = -1
    return
endif
call ncdinq (fid, dimId, dimName, jm, rc)
if (rc .NE. 0) then
    rc = -1
    return
endif

```

```

dimId = ncdid (fid, 'lev', rc)
if (rc .NE. 0) then
  rc = -1
  return
endif
call ncdinq (fid, dimId, dimName, km, rc)
if (rc .NE. 0) then
  rc = -1
  return
endif

dimId = ncdid (fid, 'time', rc)
if (rc .NE. 0) then
  rc = -1
  return
endif
call ncdinq (fid, dimId, dimName, lm, rc)
if (rc .NE. 0) then
  rc = -1
  return
endif

call ncinq (fid, nDims, nvars, ngatts, dimId, rc)
nvars = nvars - nDims
if (rc .NE. 0) then
  rc = -1
  return
endif

return
end

!-----!
!      NASA/GSFC, Data Assimilation Office, Code 910.3, GEOS/DAS      !
!-----!

```

## 2.6 Gfio\_Inquire — Get information about a GFIO file. (Source File: *gfio.F*)

This routine is used to get as much information as possible about the contents of a GFIO file. The file handle (fid) is passed in and detailed information about dimensions and variables are returned to the application. A simpler inquire routine for dimension information is Gfio\_DimInquire.

INTERFACE:

```

subroutine Gfio_Inquire ( fid, im, jm, km, lm, nvars,
&                           title, source, contact, amiss,
&                           lon, lat, levs, levunits,
&                           yyyyymmdd, hhmmss, timinc,
&                           vname, vtitle, vunits, kmvar,
&                           valid_range , packing_range, rc)

```

*USES:*

```

Implicit NONE
include 'netcdf.inc'
include 'gfoio.h'

```

*INPUT PARAMETERS:*

integer	fid	! ----- Global Metadata ----- ! File handle from GFIO_open
---------	-----	---

*INPUT/OUTPUT PARAMETERS:*

integer	im	! size of longitudinal dimension
integer	jm	! size of latitudinal dimension
integer	km	! size of vertical dimension
integer	lm	! size of time dimension
		! On input, (im,jm,km,lm) contains the ! size of arrays (lon,lat,lev,yyyyymmdd) ! as declared in the calling program.
		! On output, (im,jm,km,lm) contains the ! size of the coordinate variables ! (lon,lat,lev,yyyyymmdd) on file.
integer	nvars	! number of variables on file

*OUTPUT PARAMETERS:*

character(*)	title	! Title of the data set
character(*)	source	! Where it came from
character(*)	contact	! Who to contact about the data set
real	amiss	! Missing value
! ----- Dimension Metadata -----		
real	lon(im)	! longitude (in degrees north) of center ! of gridbox
real	lat(jm)	! longitude (in degrees north) of center ! of gridbox
real	levs(km)	! Level (units given by levunits) of ! center of gridbox
character(*)	levunits	! units of level dimension, e.g., ! "hPa", "sigma_level"
integer	yyyyymmdd(lm)	! Year-month-day on file

```

integer          hhmmss(lm)    ! Hour-minute-second on file
integer          timinc       ! Time increment. !RL

                                ! ----- Variable Metadata -----
integer          nvars        ! number of variables on file
character*(*)   vname(nvars)  ! variable short name, e.g., "hght"
character*(*)   vtitle(nvars)! variable long name, e.g.,
                            !      "Geopotential Height"
character*(*)   vunits(nvars)! variable units, e.g., "meter/second"
integer          kmvar(nvars)! number of levels for variable; it can
                            ! either be 0 (2-D fields) or equal to km
real            valid_range(2)! optional variable valid range; set to
                            !      "amiss" if not known.

                                ! ----- Packing Metadata -----
real            packing_range(2)! optional variable range used
                                ! for 16-bit packing. If packing was not
                                ! used thean packing_range(1:2) = amiss.
                                ! NOTE: all unpacking is done transparently
                                !      by Gfio_GetVar(). The packing_range
                                ! provided here is for allowing the user
                                ! to determine whether packing has been
                                ! performed.

integer          rc           ! Error return code:
                                !   rc = 0    all is well
                                !   rc = -1   too many files open
                                !   rc = -2   time does not match
                                !   rc = -3   km does not match file
                                !   rc = -4   im or jm does not match file
                                !   rc = -7   error in ncopen
                                !   rc = -8   error in ncdid
                                !   rc = -9   vname strings not big enough

```

## REVISION HISTORY:

1998.07.02	Lucchesi	Initial interface design.
1998.07.17	Lucchesi	Initial coding.

## CONTENTS:

---

! Local Variables

```

integer vid(nvars)
integer timeId, latId, lonId, levId
integer nDims, recdim, ngatts
integer varType, nvDims, vDims(MAXVDIMS), nvAtts

```

```
integer yyyyymmdd_beg, hhmmss_beg, hour, min, sec
integer start1D, minutes(lm), iret
character*8 strBuf
character*MAXCHR dimName
integer dimSize
integer i

! REAL*4 variables for 32 bit input from netCDF file.

real*4 lon_32(im), lat_32(jm), levs_32(km)
real*4 amiss_32

! Make NetCDF errors non-fatal, but issue warning messages.

call ncpopt(NCVERBOS)

! Check length of vname string

if (LEN(vname(1)) .lt. MAXNCNAM) then
    print *, 'GFIO_Inquire: length of vname array must be at least ',
    .          MAXNCNAM, ' bytes.'
    rc = -9
    return
endif

! Check to make sure max string lengths are large enough. NetCDF defines
! MAXNCNAM, but it can't be used in a character*MAXNCNAM statement.
! MAXCHR is a CPP define in the gfio.h file.

if (MAXCHR .LT. MAXNCNAM) then
    print *, 'GFIO_GetVar warning: MAXNCNAM is larger than ',
    .          'dimName array size.'
endif

! Dimension error checking.

lonId = ncidid (fid, 'lon', rc)
if (rc .NE. 0) then
    rc = -4
    return
endif
call ncinq (fid, lonId, dimName, dimSize, rc)
if (rc .NE. 0) then
    rc = -4
    return
endif
if (dimSize .ne. im) then
    rc = -4
```

```
        return
    endif

latId = ncdid (fid, 'lat', rc)
if (rc .NE. 0) then
    rc = -4
    return
endif
call ncdinq (fid, latId, dimName, dimSize, rc)
if (rc .NE. 0) then
    rc = -4
    return
endif
if (dimSize .ne. jm) then
    rc = -4
    return
endif

levId = ncdid (fid, 'lev', rc)
if (rc .NE. 0) then
    rc = -3
    return
endif
call ncdinq (fid, levId, dimName, dimSize, rc)
if (rc .NE. 0) then
    rc = -3
    return
endif
if (km .ne. dimSize) then
    rc = -3
    return
endif

timeId = ncdid (fid, 'time', rc)
if (rc .NE. 0) then
    rc = -3
    return
endif
call ncdinq (fid, timeId, dimName, dimSize, rc)
if (rc .NE. 0) then
    rc = -3
    return
endif
if (lm .ne. dimSize) then
    rc = -3
    return
endif
```

```

call ncinq (fid,nDims,dimSize,ngatts,reccdim,iret)
if (iret .NE. 0) then
  rc = -1
  return
endif
dimSize = dimSize - nDims
if (dimSize .NE. nvars) then
  rc = -3
  return
endif

start1D=1

! Get dimension values

! Dimension values in netCDF file are 32 bit.

call ncvgt (fid,lonId,start1D,im,lon_32,rc)
do i=1,im
  lon(i)=lon_32(i)
enddo
call ncvgt (fid,latId,start1D,jm,lat_32,rc)
do i=1,jm
  lat(i)=lat_32(i)
enddo
call ncvgt (fid,levId,start1D,km,levs_32,rc)
do i=1,km
  levs(i)=levs_32(i)
enddo
call ncvgt (fid,timeId,start1D,lm,minutes,rc)

! Get dimension attributes.

call ncagtc (fid,levid,'units',levunits,LEN(levunits),rc)
if (rc .NE. 0) then
  print *, 'GFTIO_Inquire: Error getting levunits.'
endif

call ncagt (fid,timeid,'time_increment',timinc,rc)
if (rc .NE. 0) then
  print *, 'GFTIO_Inquire: Error getting time_increment.'
endif

call ncagt (fid,timeid,'begin_date',yyyymmdd_beg,rc)
if (rc .NE. 0) then
  print *, 'GFTIO_Inquire: Error getting begin_date.'
endif

```

```

call ncagt (fid,timeid,'begin_time',hhmmss_beg,rc)
if (rc .NE. 0) then
  print *, 'GFIO_Inquire: Error getting begin_time.'
endif

! Calculate and load YYYYMMDD and HHMMSS values.

do i=1,lm
  call GetDate (yyyymmdd_beg,hhmmss_beg,minutes(i)*60,
.           yyyymmdd(i),hhmmss(i),rc)
enddo

! Get global attributes

call ncagtc (fid,NCGLOBAL,'Title',title,LEN(title),rc)
if (rc .NE. 0) then
  print *, 'GFIO_Inquire: Error reading global attribute Title'
endif
call ncagtc (fid,NCGLOBAL,'Source',source,LEN(source),rc)
if (rc .NE. 0) then
  print *, 'GFIO_Inquire: Error reading global attribute Source'
endif
call ncagtc (fid,NCGLOBAL,'Contact',contact,LEN(contact),rc)
if (rc .NE. 0) then
  print *, 'GFIO_Inquire: Error reading global attribute Contact'
endif

! Get variable information.

do i=1,nvars
  vid(i)=ndims+i
  call ncvinq (fid,vid(i),vname(i),varType,nvDims,vDims,
.           nvAtts,rc)
  if (rc .NE. 0) then
    print *, 'GFIO_Inquire: error inquiring about variable ',
.           vid(i)
  endif
  if (nvDims .EQ. 3) then
    kmvar(i)=0
  else
    kmvar(i)=km
  endif
  call ncagtc (fid, vid(i), 'long_name', vttitle(i),
.           LEN(vttitle(i)), rc)
  if (rc .NE. 0) then
    print *, 'GFIO_Inquire: error getting title for ',
.           vname(i)
  endif

```

```

    call ncagtc (fid, vid(i), 'units', vunits(i), LEN(vunits(i)),
    .           rc)
    if (rc .NE. 0) then
        print *, 'GFIO_Inquire: error getting units for ',
    .           vname(i)
    endif
enddo

! Get missing value.  GFIO forces this to be the same for all variables.

call ncagt (fid, ndims+1,'missing_value',amiss_32,rc)
if (rc .NE. 0) then
    print *, 'GFIO_Inquire: error getting fill value for ',
    .           vname(i)
endif
amiss = amiss_32

rc=0
return
end

!-----
!      NASA/GSFC, Data Assimilation Office, Code 910.3, GEOS/DAS      !
!-----
```

---

## 2.7 Gfio\_Close — Closes file (Source File: *gfio.F*)

This routine is used to close an open GFIO stream.

**INTERFACE:**

```
subroutine Gfio_Close ( fid, rc )
```

**USES:**

```
Implicit NONE
include 'netcdf.inc'
include 'gfio.h'
```

**INPUT PARAMETERS:**

```
integer          fid          ! File handle
```

**OUTPUT PARAMETERS:**

```
integer          rc          ! Error return code:
                           !   rc = 0    all is well
                           !   rc = -1   invalid fid
```

**REVISION HISTORY:**

1997.10.13	da Silva/Lucchesi	Initial interface design.
1998.03.30	Lucchesi	Documentation expanded. Clean-up of code. Added rc.

**CONTENTS:**


---

```

integer iret,i

! Make NetCDF errors non-fatal, but issue warning messages.

call ncpopt(NCVERBOS)

call ncclos (fid, iret)
if (iret .NE. 0) then
  rc = -1
  return
endif

rc = 0
return
end

!-----!
!      NASA/GSFC, Data Assimilation Office, Code 910.3, GEOS/DAS      !
!-----!
```

---

**2.8 Gfio\_PutIntAtt — Write a user-defined integer attribute (Source File: *gfio.F*)**

This routine allows the user to define an integer attribute in an open GFIO file.

**INTERFACE:**

```
subroutine Gfio_PutIntAtt ( fid, name, count, buf, rc )
```

**USES:**

```
Implicit NONE
include 'netcdf.inc'
include 'gfio.h'
```

**INPUT PARAMETERS:**

```
integer      fid          ! File handle
character*(*) name        ! Name of attribute
integer      count        ! Number of integers to write
integer      buf(count)   ! Buffer with integer values
```

*OUTPUT PARAMETERS:*

```
integer          rc      ! Error return code:
                  !   rc = 0    all is well
                  !   rc = -1   invalid fid or missing dimension
```

*REVISION HISTORY:*

1998.07.30	Lucchesi	Initial interface design.
1998.07.30	Lucchesi	Initial coding.

*CONTENTS:*


---

```
!-----
integer rc2

call ncredf ( fid, rc2 )
call ncapt ( fid, NCGLOBAL, name, NCLONG, count, buf, rc )
call ncendf ( fid, rc2 )

return
end

!-----
!           NASA/GSFC, Data Assimilation Office, Code 910.3, GEOS/DAS      !
!-----
```

---

**2.9 Gfio\_PutRealAtt — Write a user-defined real attribute (Source File: *gfio.F*)**

This routine allows the user to define a real attribute in an open GFIO file.

*INTERFACE:*

```
subroutine Gfio_PutRealAtt ( fid, name, count, buf, rc )
```

*USES:*

```
Implicit NONE
include 'netcdf.inc'
include 'gfio.h'
```

*INPUT PARAMETERS:*

```
integer          fid      ! File handle
character*(*)  name     ! Name of attribute
integer          count    ! Number of integers to write
real            buf(count) ! Buffer with real values
```

*OUTPUT PARAMETERS:*

```
integer          rc      ! Error return code:  
                  !   rc = 0    all is well  
                  !   rc = -1   invalid fid or missing dimension
```

*REVISION HISTORY:*

1998.07.30	Lucchesi	Initial interface design.
1998.07.30	Lucchesi	Initial coding.

*CONTENTS:*


---

```
!-----  
  
integer rc2  
  
call ncredf ( fid, rc2 )  
call ncapt ( fid, NCGLOBAL, name, NCFLOAT, count, buf, rc )  
call ncendf ( fid, rc2 )  
  
return  
end  
  
!  
!           NASA/GSFC, Data Assimilation Office, Code 910.3, GEOS/DAS      !  
!
```

---

**2.10 Gfio\_PutCharAtt — Write a user-defined character attribute (Source File: *gfio.F*)**

This routine allows the user to define a character (string) attribute in an open GFIO file.

*INTERFACE:*

```
subroutine Gfio_PutCharAtt ( fid, name, count, buf, rc )
```

*USES:*

```
Implicit NONE  
include 'netcdf.inc'  
include 'gfio.h'
```

*INPUT PARAMETERS:*

```
integer          fid      ! File handle  
character(*)  name      ! Name of attribute  
integer          count     ! Number of characters to write  
character        buf(count) ! Buffer containing string
```

*OUTPUT PARAMETERS:*

```
integer      rc      ! Error return code:  
            !   rc = 0    all is well  
            !   rc = -1   invalid fid or missing dimension
```

*REVISION HISTORY:*

1998.07.30	Lucchesi	Initial interface design.
1998.07.30	Lucchesi	Initial coding.

*CONTENTS:*


---

```
!-----  
  
integer rc2  
  
call ncredf ( fid, rc2 )  
call ncaptc ( fid, NCGLOBAL, name, NCCHAR, count, buf, rc )  
call ncendf ( fid, rc2 )  
  
return  
end  
  
!-----  
!      NASA/GSFC, Data Assimilation Office, Code 910.3, GEOS/DAS      !  
!-----
```

---

**2.11 Gfio\_GetAttNames - Get global attribute names (Source File: *gfio.F*)**

This routine allows the user to get the names of global attributes.

*INTERFACE:*

```
subroutine Gfio_GetAttNames ( fid, ngatts, aname, rc )
```

*USES:*

```
Implicit NONE  
include 'netcdf.inc'  
include 'gfio.h'
```

*INPUT PARAMETERS:*

```
integer      fid      ! File handle
```

*INPUT/OUTPUT PARAMETERS:*

```
integer      ngatts      ! Expected number of attributes (input)  
                      ! Actual number of attributes (output if rc=-2)
```

*OUTPUT PARAMETERS:*

```

character*(*)  fname(ngatts) ! Array of attribute names
integer         rc          ! Error return code:
                      !   rc = 0  all is well
                      !   rc = -1 invalid fid
                      !   rc = -2 ngatts did not match file
                      !   rc = -3 fname strings not long enough
                      !   rc = -4 error reading attribute name

```

*REVISION HISTORY:*

1998.08.05	Lucchesi	Initial interface design.
1998.08.05	Lucchesi	Initial coding.

*CONTENTS:*


---

```

integer ngattsFile, i, iret
integer nDims, dimSize, recDim

! Make NetCDF errors non-fatal, but issue warning messages.

call ncpopt(NCVERBOS)

! Check number of attributes against file

call ncinq (fid,nDims,dimSize,ngattsFile,recdim,iret)
if (iret .NE. 0) then
  rc = -1
  return
endif
if (ngattsFile .NE. ngatts) then
  rc = -2
  ngatts = ngattsFile
  return
endif

! Check length of vname string

if (LEN(fname(1)) .lt. MAXNCNAM) then
  print *, 'GFIO_GetAttNames: length of fname array must be at ',
  .      'least ',MAXNCNAM,' bytes.'
  rc = -3
  return
endif

do i=1,ngatts
  call ncanam (fid, NCGLOBAL, i, fname(i), iret)

```

```

        if (iret .NE. 0) then
          rc = -4
          return
        endif
      enddo

      return
    end

!-----  

!       NASA/GSFC, Data Assimilation Office, Code 910.3, GEOS/DAS      !
!-----  


```

---

## **2.12 Gfio\_AttInquire — Get information about an attribute (Source File: *gfio.F*)**

This routine allows the user to get information about a global attribute of an open GFIO file. This is most useful for determining the number of values stored in an attribute.

### **INTERFACE:**

```
subroutine Gfio_AttInquire ( fid, name, type, count, rc )
```

### **USES:**

```
Implicit NONE
include 'netcdf.inc'
include 'gfio.h'
```

### **INPUT PARAMETERS:**

```
integer         fid          ! File handle
character*(*)  name         ! Name of attribute
```

### **OUTPUT PARAMETERS:**

```
integer         type         ! NetCDF Code for attribute type
integer         count        ! Number of items (length of array)
integer         rc           ! Error return code:
                           !   rc = 0    all is well
                           !   rc = -1   invalid fid or attribute name
```

### **REVISION HISTORY:**

1998.07.30	Lucchesi	Initial interface design.
1998.07.30	Lucchesi	Initial coding.

### **CONTENTS:**

```
!-----
call ncainq (fid, NCGLOBAL, name, type, count, rc)

return
end

!-----  

!      NASA/GSFC, Data Assimilation Office, Code 910.3, GEOS/DAS      !
!-----
```

---

## 2.13 Gfio\_GetIntAtt — Write a user-defined integer attribute (Source File: *gfio.F*)

This routine allows the user to define an integer attribute in an open GFIO file.

### INTERFACE:

```
subroutine Gfio_GetIntAtt ( fid, name, count, buf, rc )
```

### USES:

```
Implicit NONE
include 'netcdf.inc'
include 'gfio.h'
```

### INPUT PARAMETERS:

```
integer      fid          ! File handle
character*(*) name        ! Name of attribute
```

### INPUT/OUTPUT PARAMETERS:

```
integer      count       ! On input: Number of items in attribute
                  ! On output: If rc = -1, count will contain
                  !           the correct number of attributes
```

### OUTPUT PARAMETERS:

```
integer      buf(count) ! Buffer with integer values
integer      rc         ! Error return code:
                  !   rc = 0    all is well
                  !   rc = -1   invalid count
                  !   rc = -2   type mismatch
```

### REVISION HISTORY:

1998.07.30	Lucchesi	Initial interface design.
1998.07.30	Lucchesi	Initial coding.

## CONTENTS:

```
!-----
integer length, type

call ncainq (fid, NCGLOBAL, name, type, length, rc)
if ( count .NE. length ) then
  rc = -1
  count = length
  return
else
  count = length
endif
if ( type .NE. NCLONG) then
  rc = -2
  return
endif

call ncagt ( fid, NCGLOBAL, name, buf, rc )

return
end

!-----  

!      NASA/GSFC, Data Assimilation Office, Code 910.3, GEOS/DAS      !
!-----
```

---

**2.14 Gfio\_GetRealAtt — Read a user-defined real attribute (Source File: *gfio.F*)**

This routine allows the user to read a real attribute in an open GFIO file.

## INTERFACE:

```
subroutine Gfio_GetRealAtt ( fid, name, count, buf, rc )
```

## USES:

```
Implicit NONE
include 'netcdf.inc'
include 'gfio.h'
```

## INPUT PARAMETERS:

```
integer          fid          ! File handle
character(*)   name         ! Name of attribute
```

## INPUT/OUTPUT PARAMETERS:

```
integer      count      ! On input: Number of items in attribute
              ! On output: If rc = -1, count will contain
              !           the correct number of attributes
```

*OUTPUT PARAMETERS:*

```
real         buf(count) ! Buffer with integer values
integer      rc        ! Error return code:
              !   rc = 0    all is well
              !   rc = -1   invalid count
              !   rc = -2   type mismatch
```

*REVISION HISTORY:*

1998.07.30	Lucchesi	Initial interface design.
1998.07.30	Lucchesi	Initial coding.

*CONTENTS:*


---

```
!-----
integer length, type

call ncainq (fid, NCGLOBAL, name, type, length, rc)
if ( count .NE. length ) then
  rc = -1
  count = length
  return
else
  count = length
endif
if ( type .NE. NCFLOAT) then
  rc = -2
  return
endif

call ncagt ( fid, NCGLOBAL, name, buf, rc )

return
end

!-----
```

---

```
!           NASA/GSFC, Data Assimilation Office, Code 910.3, GEOS/DAS      !
!-----
```

---

**2.15 Gfio\_GetCharAtt — Read a user-defined character attribute (Source File: *gfio.F*)**

This routine allows the user to read a character attribute in an open GFIO file.

**INTERFACE:**

```
subroutine Gfio_GetCharAtt ( fid, name, count, buf, rc )
```

**USES:**

```
Implicit NONE
include 'netcdf.inc'
include 'gfio.h'
```

**INPUT PARAMETERS:**

```
integer      fid          ! File handle
character*(*) name        ! Name of attribute
```

**INPUT/OUTPUT PARAMETERS:**

```
integer      count        ! On input: Number of items in attribute
                      ! On output: If rc = -1, count will contain
                      !           the correct number of attributes
```

**OUTPUT PARAMETERS:**

```
character    buf(count) ! Buffer with integer values
integer      rc          ! Error return code:
                      !   rc = 0    all is well
                      !   rc = -1   invalid count
                      !   rc = -2   type mismatch
```

**REVISION HISTORY:**

1998.07.30	Lucchesi	Initial interface design.
1998.07.30	Lucchesi	Initial coding.

**CONTENTS:**


---

```
integer length, type

call ncainq (fid, NCGLOBAL, name, type, length, rc)
if ( count .NE. length ) then
  rc = -1
  count = length
  return
else
  count = length
endif
if ( type .NE. NCCHAR) then
  rc = -2
  return
endif
```

```
call ncagtc ( fid, NCGLOBAL, name, buf, count, rc )  
  
return  
end
```